Supplementary Information with:

Assembly dynamics of microtubules at molecular resolution

Jacob W.J. Kerssemakers[1,2], E. Laura Munteanu[1], Liedewij Laan[1], Tim L. Noetzel[2], Marcel E. Janson[1,3], and Marileen Dogterom[1]

[1]*FOM Institute AMOLF, Kruislaan 407, 1098 SJ Amsterdam, The Netherlands*

[2]*MPI-CBG Dresden, Pfotenhauerstrasse 108, 01307 Dresden, Germany*

[3]*University of Pennsylvania, Dept. of Cell & Developmental Biology, 421 Curie Blvd., Philadelphia, PA 19104-6058, USA*

Supplementary Methods 3

Step fitting algorithm

Introduction

Evaluating possible step-like processes in otherwise noisy data sets is a returning problem in many biophysics studies. With non-constant step sizes and small numbers of step events, a classical method like evaluating pairwise distance distribution functions does not suffice. We developed a simple, practical algorithm that allows us to distinguish pronounced step-like behavior from gradual non-stepped growth, and to return the size distribution of the steps that are distinguishable from the noise. The sole assumption we make is that the original data is a step train with steps of varying size and duration, hidden in Gaussian noise with RMS amplitude σ.

Summary of the algorithm

Our step-fitting algorithm involves 3 steps:

1. Finding steps (Fig. C1a):

The algorithm starts by fitting a single large step to the data, finding the size and location of this first step based on a calculation of the Chi-squared. Subsequent steps are found by fitting new steps to the plateaus of the previous ones, each time selecting the most prominent one first. This eventually leads to a series of 'best' fits that differ only by one step. The fits with a very low number of steps are likely to underestimate or 'underfit' the real number of steps in the data, whereas the small steps that are added in the last iterations will merely be fitting the noise, thereby 'overfitting' the data.

2. Evaluating the quality of the step fits:

Each best fit in the series is compared to a 'counter fit' that has an equal number of steps as the original one but with step locations in between the step locations found by the best fit (Fig. C1b). We define a 'step-indicator' *S* as the ratio between the Chi-squared of the counter fit and the Chi-squared of the best fit. When the number of steps in the best fit is very close to the real number of steps in the data, the value of *S* will be large (Fig. C1c). If however the data are severely under- or overfitted, or when the data consist of gradual non-stepped growth, the value for *S* will be close to 1.

3. Finding step distributions:

To construct a histogram of the step sizes, an 'optimal' fit (the one representing best the real steps in the data) has to be chosen. Ideally this is the one with the number of steps that produces the highest value of *S*. In practice, we usually choose a fit that appears to slightly overfit the data (see Figs. C2 and C3).

This procedure allows us to i) distinguish step-like growth from gradual non-stepped increases in length in a quantitative way, and ii) identify and quantify the steps that are distinguishable from the noise. The individual steps do not have to be equal in size or duration, and no a-priori assumptions about the signal to noise ratio are necessary. Of course, if the underlying steps are small compared to the noise, the algorithm will not be able to reliably distinguish a train of steps from linear non-stepped growth, which will manifest itself through a low value of *S*. If, as is the case for tubulin growth, the data consist of combinations of steps that are large and small compared to the noise, the algorithm will ensure that we find the sizes of the large ones. However, the optimal fit that we find based on the size of the large steps, will also put arbitrary steps on the rest of the data.

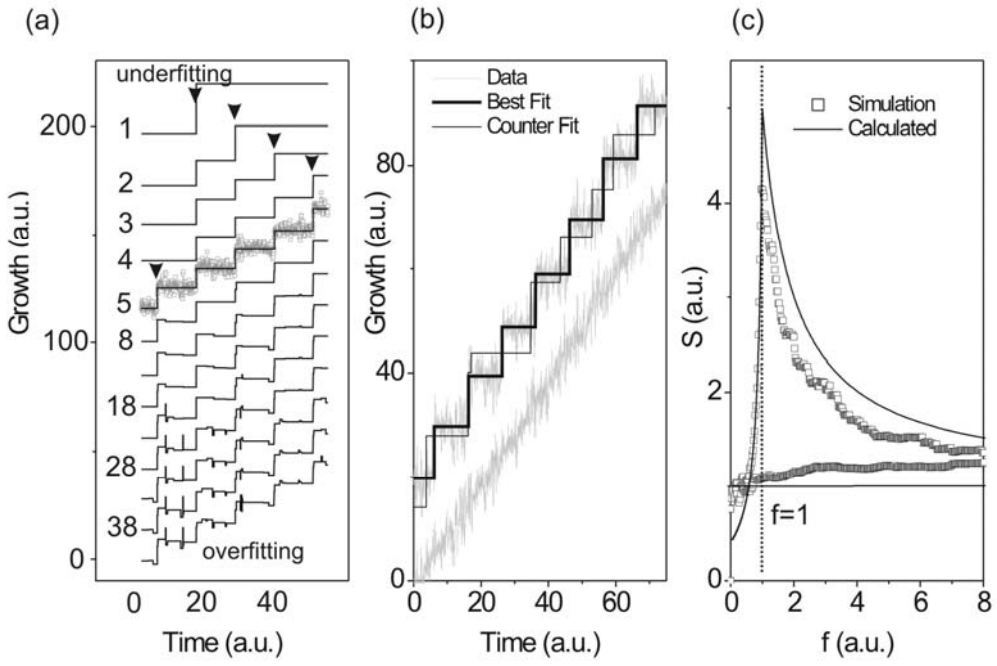

Figure C1. *Step fitting procedure. a) Iterations of the step-fitting algorithm on a simulated, noisy track of stepped data (step size 10 nm, RMS noise 2.5 nm). Curves are shifted vertically for clarity. The arrowheads point to every new step that is added to the fit. Underfitting means that significant steps in the data are not yet located, while overfitting means that*

merely noise is fitted. b) A "best" fit (thick line) to noisy steps (step size 10 nm, RMS noise 2.5 nm) together with a "counter" fit (thin line, see text). The quality of these two fits differs strongly for a stepped signal, while for linear noisy growth (lower curve), the location of any step is arbitrary and the quality is equal. c) Simulation result and calculation of the quality ratio S of best fit and counter fit, plotted vs. the relative number of fitted steps f. Upper curves are for a noisy stepped signal. Lower curves are for a noisy linear signal. S only peaks sharply if there are steps present, and if the correct number of steps is fitted (f=1).

Below, the different steps of the algorithm are explained in more detail together with some theoretical estimates of what to expect for relatively simple data sets.

Finding Steps

In Figure C1, left panel, the fitting procedure for a simulated, noisy step train is shown. The algorithm starts by fitting a single step to this data with a location and size that gives the lowest residual Chi^2. For a data set of N points with Gaussian noise σ that contains no actual steps, this will on average produce a step in the middle with a (stochastic) step size of $\sigma^2/2N$. This will also be the amount by which, on average, the residual Chi^2 will be lowered with respect to a "zero-step" fit. If, however, there is a true step hidden in the data, the best fit will strongly favor the location of this step, as this will lower the Chi^2 by an amount $\sigma^2/2N+\Delta^2/4$, where Δ is the fitted step size. This will still be true when there are more steps in the data, although the single-step fit will of course not yield a perfect fit yet. The best single-step fit for the simulated data in Figure C1 is shown as the upper curve, shifted upwards for clarity purposes. Next, on each of the resulting two plateaus of the first step, a new step fit is performed with the step locations again on the locations that give the lowest residual Chi^2. These new steps each come with a fitted step size Δ and a step 'window': the number of data

points N_w between the neighboring locations, left and right of the fitted step location. Since we want to add only one step per iteration, we keep only the step with the largest value of the parameter $\xi = \Delta\sqrt{N_w}$ (see arrowheads in Figure C1). This ensures that the most prominent features, with large step size Δ or large window size N_w are selected first. The process is repeated, producing increasingly finer step fits to the data, until only a few data-points are left per step. Once found, the step-locations do not change anymore, although the associated step sizes continue to change as they depend on the location of the neighboring steps (see Figure C1). Note that this procedure will eventually 'overfit' the data since the last iterations will merely be fitting the noise. In the next section we describe a procedure to evaluate the quality of the fits, which will help us find the optimal fit.

Evaluating Step Fits

In practical cases, many data tracks do show significant changes, but not necessarily step-like ones. One important aim of our procedure is to make a distinction between stepped and non-stepped growth (or shrinkage). With non-stepped growth we mean a gradually changing length, without sudden, statistically significant transitions. Imagine a simple, linear growth signal immersed in noise. If we would apply the above procedure, we would again find increasingly finer step fits, but now the locations of the steps would be more or less arbitrary. This means that if we would deliberately place the steps on any other set of locations, the fit would be nearly as good. In contrast, if the same growth would consist of a train of steps with size Δ and duration N_w, misplacing the steps would in general yield a much worse fit. This is especially the case at the iteration of the step-fitting algorithm that just matches all the steps in the data: while the 'best' fit would only leave a residual Chi^2 of order σ^2, an

arbitrary 'mis' fit would on average yield $\sigma^2+\Delta^2/4$. We make use of this property to refine the algorithm in the following way: at each iteration a secondary, 'counter' fit is performed by doing best-step fits on the plateaus in between the fitted steps and, after that, rejecting the original step locations (see figure C1, middle panel). The result is a full set of step locations that are all in between the best-fit locations, albeit sometimes close to the original one (Figure C1).

This counter fitting gives the following information:

i) If the best fit is at its optimal iteration, all significant steps are just covered. In contrast, all step locations found with the counter fit are completely misplaced: the quality of the best fit and counter fit differ strongly, as outlined before.

ii) In the case of severe underfitting, when there are still many true steps left in all fitted plateaus, the quality of the counter fit will not differ much from the best fit, as in both cases the fits will locate significant steps.

iii) In the case of severe overfitting, both counter fit and best fit are likely to find merely noise excursions, and will also exhibit a similar quality.

iv) If the signal consists of non-stepped growth that is close to linear, any choice of step locations is arbitrary. Then, the quality of the best fit and the counter fit also only differ by statistical variation.

These situations can be quantified by defining S as the ratio of the residual Chi-squares: $S=\mathrm{Chi}^2_{\text{counter fit}}/\mathrm{Chi}^2_{\text{best fit}}$. For the last three cases above, S would be close to 1. However, in case i), at precisely the optimal number of fitted steps, the ratio of the Chi-squares of counter fit and best fit is roughly given by $S=(\Delta^2/4+\sigma^2)/\sigma^2$ or $1+\Delta^2/4\sigma^2$ for a step train containing only regular steps and noise. Therefore, S is a

'step-indicator': it only departs from unity when a) anything steps in a statistically significant way and b) if these steps are correctly fitted. For a regular step train, *S* can also be calculated for all non-optimal cases discussed above: slight underfitting, slight overfitting, and heavy overfitting. In essence, around the peak value any missed or over-fitted step lowers the ratio by an amount $\Delta^2/4$, weighted by how large the window size of this step is compared to the full duration of the whole dataset. At very few data points per step window (i.e. large numbers of fitted steps), the statistical advantage of the best fit over the counter fit becomes significant, and *S* rises in a so-called noise tail. For a fit of N_f steps to a curve consisting of N_r real steps, we can thus approximate *S* as a function of $f=N_f/N_r$:

$$S = \frac{1 + P - \frac{P}{3}(1-f)}{1 + 2P(1-f)} \qquad \text{for } f<1 \text{ (underfitting), and} \qquad [1]$$

$$S = \frac{P + f}{1 + (f-1)(1-\frac{1}{2n})} \qquad \text{for } f>1 \text{ (overfitting)}$$

where $P \equiv \left[\Delta / 2\sigma\right]^2$ is the peak value and $n=N_0/N_r$ is the number of samples points per step event. In contrast, comparable growth (with the same average elongation rate) that is linear or that consists of very small steps that are drowned in the noise, yields:

$$S = \frac{P+3}{P+3(1-\frac{1}{2N})} \qquad [2]$$

In the right panel of Figure C1, the calculated results are compared with the output of the algorithm on data shown in the middle panel. For the step train, *S* indeed sharply peaks at $f=1$, i.e. at the actual number of steps present. As noted before, the peak value is $1+(\Delta/2\sigma)^2$ and therefore scales quadratically with the signal/noise ratio. In contrast, there is no such peak in the curve for linear growth. We note that the method works on any signal containing steps, not just on ones with uniformly sized and spaced steps.

This makes the algorithm a powerful tool to evaluate data with no *a priori* assumptions on any step distribution that might be there.

Step distributions

In practice, a useful way to find the optimal fit and at the same time get an indication of the average step size in the data, is to plot S a function of the parameter $X=L_{tot}/N_f$, where L_{tot} is the total covered length between the beginning and the end of the data track and N_f is the number of (positive and negative) fitted steps. X is defined such that it monotonically decreases for increasingly finer step fits, and in case there are only positive (or negative) steps, it represents the average size of the fitted steps. In Figure C2, middle panel, S is plotted for similar step-less (I) and stepped (II) data sets as in Figure C1 (see left panel), but this time as a function of X. As before, curve II represents a simulated noisy step train with steps of Δ=20 nm immersed in noise σ (10 nm RMS). In addition, curve III represents steps obtained experimentally in our setup, where a trapped bead-axoneme construct was pushed away with steps of Δ~20 nm from the trap center by a piezo-controlled barrier (RMS noise σ~5 nm). For both step trains, S peaks sharply at the pre-set step size around 20 nm with the peak height scaling approximately with $[\Delta/\sigma]^2$. In contrast to this, the step-less growth only features a noise tail at small average step sizes, while S steeply drops down to unity for any larger step size.

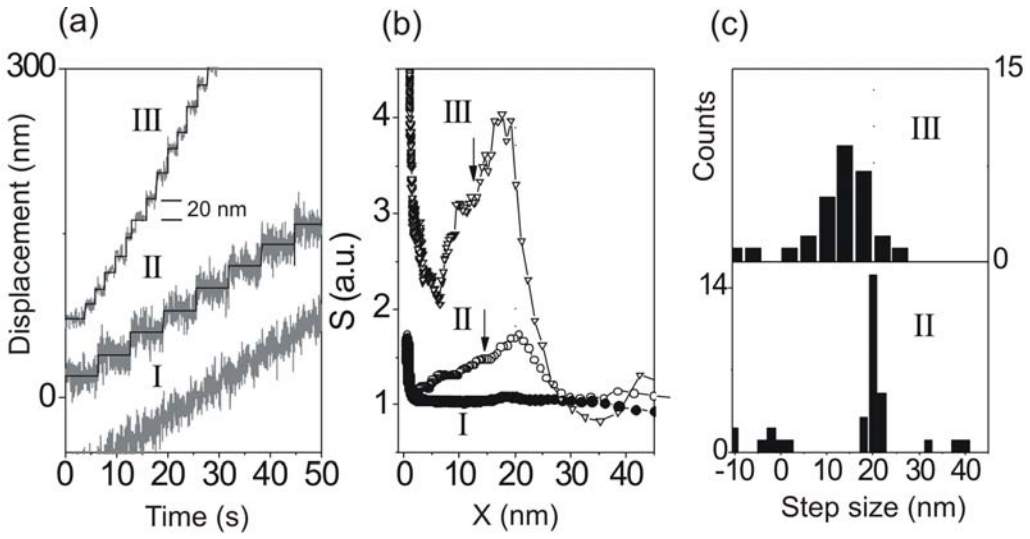

Figure C2. *Finding step distributions. a) Raw data of simulated noisy step-less growth (I), simulated 20 nm steps in 10 nm RMS noise (II), and experimental steps of ~20 nm in ~5 nm RMS noise, obtained by stepwise pushing a bead-axoneme construct away from the trap center (III). b) S as a function of X for these data (see text), showing peaks only when steps are present. Arrows indicate the step fits chosen to construct the step size histograms. c) Associated step size histograms.*

To evaluate the distribution of step sizes, one fit has to be chosen. For practical purposes, moderate overfitting, i.e. taking a point on the left side of the peak in *S* in Figure C2 (arrows), ensures that all significant steps are found and only a minor amount of small, noise-related steps are fitted extra. Then, step histograms can be made that depict the full step size distribution (see the right panel in Figure C2). We note that by plotting *S* as a function of *X* (middle panel), the resulting curves resemble the final step distributions (right panel). Thus, a rough approximation of the typical step sizes present in the data is already found by just running the step-finding algorithm and plotting *S* as a function of *X*, without any choice yet for an optimal step fit.

In Figure C3 we show the result of this procedure for the growth data with pure tubulin presented in the main text. Note that in this case a substantial fraction of the growth occurs through steps that are indistinguishable from the noise and/or linear growth. Our algorithm is designed to find the sizes of steps that are distinguishable from the noise, but also forces steps on the rest of the data. These last step sizes vary strongly when we vary the number of steps fitted to the data, but the 20-30 nm peak remains present even when we clearly start to overfit the data (see Fig. C3).

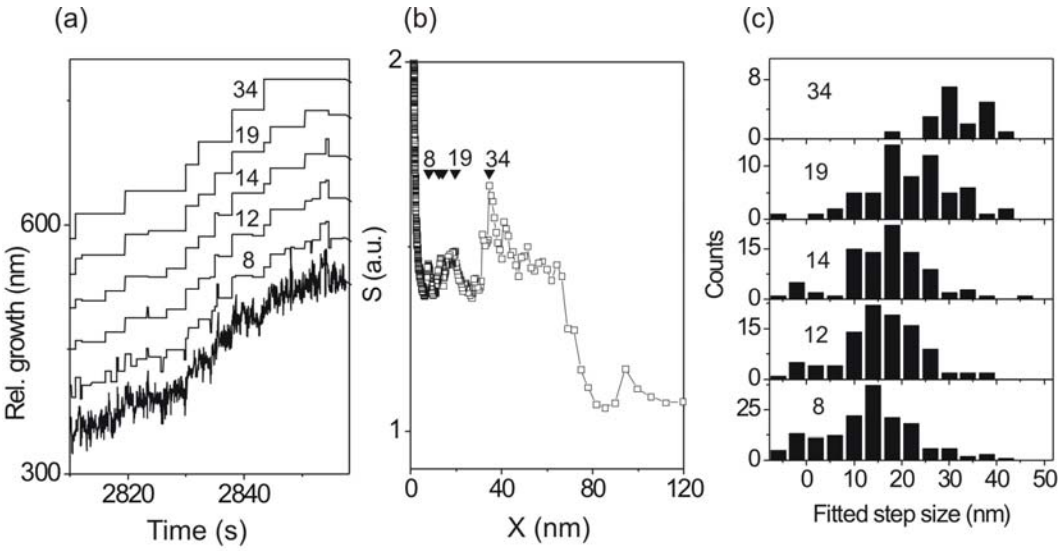

Figure C3. *Fitting experimental data for pure tubulin growth. a) Part of the raw data of a pushing MT with a series of increasingly finer step-fits. The numbers refer to the corresponding value of the parameter X. b) S as a function of X for the complete data set, showing values above 1 over a wide range of length scales. Arrowheads and numbers refer to the specific step-fits shown in a). c) Associated step-distributions. Step sizes up to 20-30 nm are found even when the data are clearly "overfitted" (for X=8). Note that overfitting mostly leads to the addition of alternating up- and down steps.*